WE CLAIM:

1.   A software architecture for a client device, the architecture comprising:

a data store for storing a predefined data structure having a first data object and a predefined first template having a first identifier value and a first field, where the first field of the

5   first template is tagged as corresponding to the first data object of the data structure;

a first application process configured to output information to a user of the client device in a format defined by the first template, where the first application process, responsive to a user command, is configured to format and send a template population request message that includes an identifier field having the first identifier value for the first template, and is further configured

10   to render the first template, when populated with data, for display to a user of the client device;

a server process within the client device, the server process being configured to receive the template population request message from an application and, responsive thereto, use the value from the template identifier value from the template population request message to retrieve a corresponding template from the data store, use the tag from the first field of the corresponding

15   template to retrieve a value from a corresponding data object of the data structure corresponding to the tag value from the corresponding template, and return the corresponding template populated with the value from the data object to the application that sent the template population request message.

20   2.   The architecture of claim 1, where:

- 95 -

the data store includes a predefined second template having a second identifier value and a first field, where the first field of the second template is tagged as corresponding to the first data object of the data structure;

the architecture further includes a second application process configured to output

5   information to the user of the client device in a format defined by the second template, where the second application process, responsive to a user command, is configured to format and send the template population request message having the second identifier value for the second template in the identifier field of the message, and is further configured to render the first template, when populated with data, for display to the user of the client device.

10

3.   The architecture of claim 2, where:

the data structure further comprises an XML data structure; and

the server process is configured to obtain data from the XML data structure in the data store using a Server Side Include (SSI) command.

15

4.   The architecture of claim 2, where the automatic update process for communicating with a remote server is further configured to automatically download and store the data value in the first data object of the data structure in the data store at a predetermined periodic time interval.

20

5.   The architecture of claim 4, where the periodic time intervals include times of low use by the user of the client device.

6.   The architecture of claim 4, where the periodic time intervals includes times of low use by the user of the communication link.

5   7.   The architecture of claim 4, where:

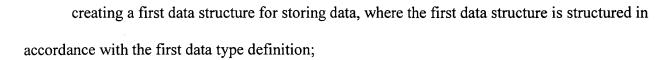the data structure further comprises an XML data structure;

the server process is configured to obtain data from the XML data structure in the data store using a Server Side Include (SSI) command; and

the automatic update process is configured to receive an XML data document containing

10   the value for the first data object, which the automatic update process is configured to parse the value for the first data object from the XML data document and insert the value for the first data object into the data structure.

8.   The architecture of claim 2, where the remote server defines the data value in the

15   first data object of the data structure in the data store according to a first predetermined data type definition and the first application process accesses the first data object of the data structure in the data store according to the first predetermined data type definition.

9.   A method for sharing data between multiple applications in a client device, the

20   method comprising the steps:

defining a first predetermined data type definition;

- 97 -

creating a first data structure for storing data, where the first data structure is structured in

accordance with the first data type definition;

storing a first data object from a first application in the first data structure, where the first

data object of the first application is structured according to the first data type definition;

5   retrieving the first data object in a second application by accessing the first data structure

according to the first predetermined data type definition; and

outputting the first data object from the second application to a user of the client device.

R1.126   16.   The method of claim 9, where:

10   the first predetermined data type defines a weather data object;

the second application includes a calendar function;   )

the step of retrieving the first data object in a second application further includes

retrieving the weather data object from the first data structure; and

the step of outputting the first data object from the second application to a user of the

15   client device further includes displaying the weather data object along with data from the

calendar function.

R1.126   16.   The method of claim 16, where the first application is a browser application for

downloading information from a remote server.

20

R1.126   17.   The method of claim 9, where:

the first predetermined data type defines an event object;

- 98 -

the second application includes a calendar function;

the step of retrieving the first data object in a second application further includes

retrieving the event data object; and

the step of outputting the first data entry from the second application to a user of the

5   client device further includes displaying the event data object along with data from the calendar

function.

ρ1.12�6   13. The method of claim 12, where the event data object corresponds to one of a

sports event and a sales event.

10
ρ1.12�6   14. A data display device, the device comprising:

a communication interface adapted for connection to a network of computing devices;

a memory device containing a first display template corresponding to a first application

and a second display template corresponding to a second application, where each of the first and

15   second display template includes an index to a first data object, where the index is defined by a

data type definition;

a microprocessor programmed to execute the first and second applications and a database

manager for providing access to a database, where the microprocessor is configured to execute

each of the first and second applications by retrieving the first and second display templates,

20   respectively, and using the index to the first data object to access the database in order to retrieve

a value for the first data object, where the microprocessor is further configured to execute an

update routine for receiving periodic data updates to the first data object via the communication

interface and storing the data updates to the first data object in the database;

a display generator for generating display data from the first and second templates and

the first data object data; and

5   a display device for receiving said display data and displaying images.

14.   The data display device of claim 13, where the communication interface further

comprises one of a universal serial bus port, an ethernet port and a modem.

10   15.   The data display device of claim 13, where the database manager further

comprises one of a SQL system, a file system and an XML parser.

16.   The data display device of claim 13, where the display generator further

comprises one of a web browser and an XML processor.

15